

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-265587
 (43)Date of publication of application : 28.09.2001

(51)Int.CI.

G06F 9/06
 G06F 17/21
 G06T 1/00
 G09C 5/00
 H04N 1/387
 H04N 7/08
 H04N 7/081

(21)Application number : 2000-074168

(22)Date of filing : 16.03.2000

(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>

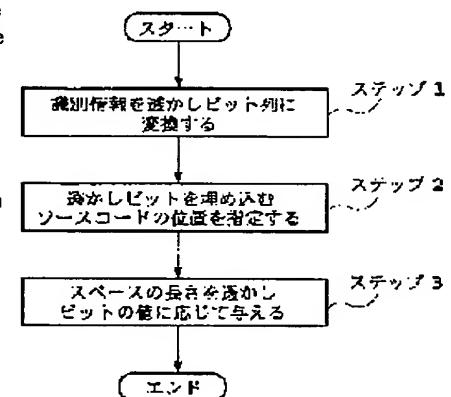
(72)Inventor : OBARA KAZUHIRO

(54) METHOD AND SYSTEM FOR EMBEDDING SOFTWARE IDENTIFICATION INFORMATION, AND RECORDING MEDIUM WITH SOFTWARE IDENTIFICATION INFORMATION EMBEDDING PROGRAM RECORDED THEREON

(57)Abstract:

PROBLEM TO BE SOLVED: To protect a program by embedding identification information such as a copyright information, an obtaining person information to the source code of a developed software so as to clarify the copyright information, the obtaining person information, etc., of the software.

SOLUTION: This invention has a step 1 for converting identification information into a watermark bit string expressed by a bit string, a step 2 for designating a position for embedding watermark bits in the source code and a step 3 for giving the space length (the number of blanks) of both of the spaces or the space length of either of the space among a space between character marks existing between a character or a mark at the position of the designated source code and the other character or mark and a row head space existing until the first character or the first mark of a line appears in accordance with the value of each bit of the watermark bit string. When the value of the watermark bit is 0, the space length is made to be 1 and when the value of the watermark bit is 1, the space length is made to be 2.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the
examiner's decision of rejection or application converted
registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of
rejection]

[Date of extinction of right]

Copyright (C) 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2001-265587
(P2001-265587A)

(43)公開日 平成13年9月28日 (2001.9.28)

(51)Int.Cl. ⁷	識別記号	F I	テマコード*(参考)
G 0 6 F 9/06	5 5 0	G 0 6 F 9/06	5 5 0 Z 5 B 0 0 9
17/21	5 9 0	17/21	5 9 0 Z 5 B 0 5 7
G 0 6 T 1/00	5 0 0	G 0 6 T 1/00	5 0 0 B 5 B 0 7 6
G 0 9 C 5/00		G 0 9 C 5/00	5 C 0 6 3
H 0 4 N 1/387		H 0 4 N 1/387	5 C 0 7 6

審査請求 未請求 請求項の数14 OL (全 8 頁) 最終頁に統く

(21)出願番号 特願2000-74168(P2000-74168)

(22)出願日 平成12年3月16日 (2000.3.16)

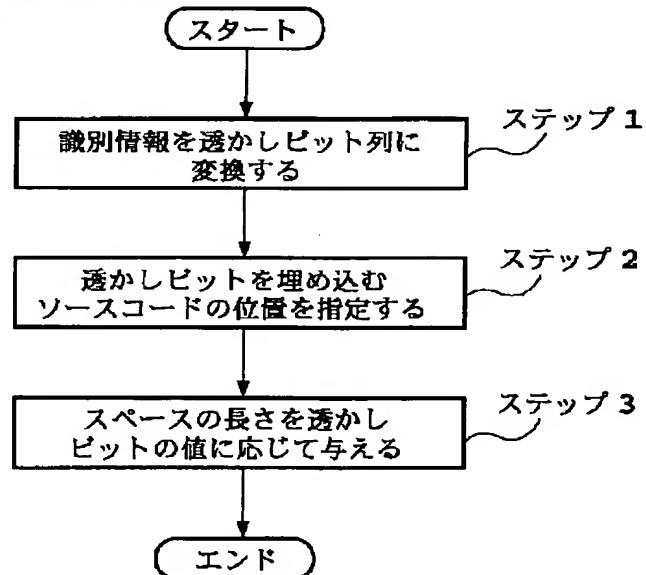
(71)出願人 000004226
日本電信電話株式会社
東京都千代田区大手町二丁目3番1号
(72)発明者 小原 和博
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内
(74)代理人 100077481
弁理士 谷 義一 (外1名)

(54)【発明の名称】 ソフトウェア識別情報埋め込み方法及びソフトウェア識別情報埋め込みシステム、並びにソフトウェア識別情報埋め込みプログラムを記録した記録媒体

(57)【要約】

【課題】 開発したソフトウェアのソースコードに著作権情報や入手者情報などの識別情報を埋め込むことで、ソフトウェアの著作権情報や入手者情報などを明らかにし、プログラムの保護を図る

【解決手段】 識別情報をビット列で表現した透かしビット列に変換するステップ1と、ソースコードの中で透かしビットを埋め込む位置を指定するステップ2と、その指定されたソースコードの位置にある、文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、透かしビット列の各ビットの値に応じて与えるステップ3とを有する。例えば、透かしビットの値が0のときにはスペース長を1とし、透かしビットの値が1のときにはスペース長を2とする。



【特許請求の範囲】

【請求項1】 プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込み方法において、前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化ステップと、前記ソースコードの同一行にある文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、各行において最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与ステップとを有することを特徴とするソフトウェア識別情報埋め込み方法。

【請求項2】 プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込み方法において、前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化ステップと、前記ソースコードの中で前記透かしビットを埋め込む位置を指定する埋め込み位置指定ステップと、前記埋め込み位置指定ステップで指定されたソースコードの位置にある、文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与ステップとを有することを特徴とするソフトウェア識別情報埋め込み方法。

【請求項3】 前記スペース長付与ステップにおいて、同一の識別情報を前記ソースコードの複数箇所に反復して埋め込むことを特徴とする請求項1または2に記載のソフトウェア識別情報埋め込み方法。

【請求項4】 前記スペース長付与ステップにおいて、前記透かしビット列の各ビットの値が0のときには前記スペース長を1とし、前記透かしビット列の各ビットの値が1のときには前記スペース長を2として、前記スペース長を与えることを特徴とする請求項1ないし3のいずれかに記載のソフトウェア識別情報埋め込み方法。

【請求項5】 前記埋め込み位置指定ステップでは、前記透かしビット列の各ビットを埋め込む位置をソースコードの行番号で指定し、該指定した行にスペースが複数個ある場合には、その埋め込みに利用するスペースの位置を選択して指定することを特徴とする請求項2に記載のソフトウェア識別情報埋め込み方法。

【請求項6】 前記埋め込み位置指定ステップでは、前記透かしビット列の各ビットを埋め込む位置をソースコードの行番号で指定し、

前記スペース長付与ステップでは、前記埋め込み位置指定ステップで指定された前記行番号に応じて、前記スペース長の与え方を変えることを特徴とする請求項2または5に記載のソフトウェア識別情報埋め込み方法。

【請求項7】 前記スペース長付与ステップでは、前記埋め込み位置指定ステップで指定された前記行番号が奇数のときには「透かしビットの値が0のときにはスペース長を1とし、透かしビットの値が1のときにはスペース長を2」とし、前記埋め込み位置指定ステップで指定された前記行番号が偶数のときには「透かしビットの値が0のときにはスペース長を2とし、透かしビットの値が1のときにはスペース長を1」として前記スペース長を付与することを特徴とする請求項6に記載のソフトウェア識別情報埋め込み方法。

【請求項8】 プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込みシステムにおいて、

前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化手段と、

前記ソースコードの同一行にある文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、各行において最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与手段とを有することを特徴とするソフトウェア識別情報埋め込みシステム。

【請求項9】 プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込みシステムにおいて、

前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化手段と、

前記ソースコードの中で前記透かしビットを埋め込む位置を指定する埋め込み位置指定手段と、

前記埋め込み位置指定手段で指定されたソースコードの位置にある、文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与手段とを有することを特徴とするソフトウェア識別情報埋め込みシステム。

【請求項10】 コンピュータによってソフトウェアのソースコードに識別情報を埋め込むためのソフトウェア識別情報埋め込みプログラムを記録した記録媒体であつて、該プログラムはコンピュータに対し、

前記識別情報をビット列で表現した透かしビット列に変換させ、

前記ソースコードの同一行にある文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、各行において最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えさせることを特徴とするソフトウェア識別情報埋め込みプログラムを記録した記録媒体。

【請求項11】 コンピュータによってソフトウェアのソースコードに識別情報を埋め込むためのソフトウェア識別情報埋め込みプログラムを記録した記録媒体であって、該プログラムはコンピュータに対し、前記識別情報をビット列で表現した透かしビット列に変換させ、

前記ソースコードの中で前記透かしビットを埋め込む位置を指定させ、前記指定されたソースコードの位置にある、文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えさせることを特徴とするソフトウェア識別情報埋め込みプログラムを記録した記録媒体。

【請求項12】 前記プログラムはコンピュータに対し、同一の識別情報を前記ソースコードの複数箇所に反復して埋め込ませることを特徴とする請求項10または11に記載のソフトウェア識別情報埋め込みプログラムを記録した記録媒体。

【請求項13】 前記プログラムはコンピュータに対し、前記透かしビット列の各ビットの値が0のときには前記スペース長を1とさせ、前記透かしビット列の各ビットの値が1のときには前記スペース長を2とさせることを特徴とする請求項10ないし12のいずれかに記載のソフトウェア識別情報埋め込みプログラムを記録した記録媒体。

【請求項14】 前記プログラムはコンピュータに対し、前記透かしビット列の各ビットを埋め込む位置を前記ソースコードの行番号で指定させ、該指定した行番号に応じて、前記スペース長の与え方を変えさせることを特徴とする請求項11に記載のソフトウェア識別情報埋め込みプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ソフトウェアのソースコードに著作権情報や入手者情報などの識別情報を埋め込むソフトウェア識別情報埋め込み方法及びその方法を実行するソフトウェア識別情報埋め込みシステム、

並びにソフトウェア識別情報埋め込みプログラムを記録した記録媒体に関する。

【0002】

【従来の技術】 近年、文字、音声、画像などのメディア情報をデジタル化して作成したコンテンツを通信ネットワークや記憶媒体（CD-ROM, フロッピー（登録商標）など）を介して流通する場合、デジタル情報はアナログ情報とは違って劣化なく複製することができるために、著作権の保護が重要な問題となっている。この問題の解決策の一つとして、著作権に関する情報をコンテンツ内に埋め込む「電子透かし」と称されている技術が色々と提案されている（参考文献1：松井甲子雄「電子透かし技術の最新動向」、情報処理、1999年2月、pp178-183）。

【0003】 しかし、これまで提案されている電子透かし技術は、メディア情報が有する冗長性を利用するものであるため、カラー画像や動画などには適用可能であるが、2値画像や、文字情報、プログラムなどには適用できなかった。

【0004】

【発明が解決しようとする課題】 本発明は、上記の点に鑑みてなされたもので、その目的は、開発したソフトウェア（プログラム）のソースコードに著作権情報や入手者情報などの識別情報を埋め込むことで、ソフトウェアの著作権情報や入手者情報などを明らかにして、プログラムの著作権保護を図ることにある。

【0005】

【課題を解決するための手段】 上記目的を達成するため、請求項1の発明は、プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込み方法において、前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化ステップと、前記ソースコードの同一行にある文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、各行において最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与ステップとを有することを特徴とする。

【0006】 上記目的を達成するため、請求項2の発明は、プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込み方法において、前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化ステップと、前記ソースコードの中で前記透かしビットを埋め込む位置を指定する埋め込み位置指定ステップと、前記埋め込み位置指定ステップで指定されたソースコードの位置にある、文字あるいは記号と他の文字ある

いは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与ステップとを有することを特徴とする。

【0007】ここで、前記スペース長付与ステップにおいて、同一の識別情報を前記ソースコードの複数箇所に反復して埋め込むことを特徴とすることができる。

【0008】また、前記スペース長付与ステップにおいて、前記透かしビット列の各ビットの値が0のときには前記スペース長を1とし、前記透かしビット列の各ビットの値が1のときには前記スペース長を2として、前記スペース長を与えることを特徴とすることができる。

【0009】また、前記埋め込み位置指定ステップでは、前記透かしビット列の各ビットを埋め込む位置をソースコードの行番号で指定し、該指定した行にスペースが複数個ある場合には、その埋め込みに利用するスペースの位置を選択して指定することを特徴とすることができる。

【0010】また、前記埋め込み位置指定ステップでは、前記透かしビット列の各ビットを埋め込む位置をソースコードの行番号で指定し、前記スペース長付与ステップでは、前記埋め込み位置指定ステップで指定された前記行番号に応じて、前記スペース長の与え方を変えることを特徴とすることができる。

【0011】また、前記スペース長付与ステップでは、前記埋め込み位置指定ステップで指定された前記行番号が奇数のときには「透かしビットの値が0のときにはスペース長を1とし、透かしビットの値が1のときにはスペース長を2」とし、前記埋め込み位置指定ステップで指定された前記行番号が偶数のときには「透かしビットの値が0のときにはスペース長を2とし、透かしビットの値が1のときにはスペース長を1」として前記スペース長を付与することを特徴とすることができる。

【0012】上記目的を達成するため、請求項8の発明は、プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込みシステムにおいて、前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化手段と、前記ソースコードの同一行にある文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、各行において最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与手段とを有することを特徴とする。

【0013】上記目的を達成するため、請求項9の発明

は、プログラムされたコンピュータによってソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込みシステムにおいて、前記識別情報をビット列で表現した透かしビット列に変換する識別情報ビット列化手段と、前記ソースコードの中で前記透かしビットを埋め込む位置を指定する埋め込み位置指定手段と、前記埋め込み位置指定手段で指定されたソースコードの位置にある、文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えるスペース長付与手段とを有することを特徴とする。

【0014】上記目的を達成するため、請求項10の発明は、コンピュータによってソフトウェアのソースコードに識別情報を埋め込むためのソフトウェア識別情報埋め込みプログラムを記録した記録媒体であって、該プログラムはコンピュータに対し、前記識別情報をビット列で表現した透かしビット列に変換させ、前記ソースコードの同一行にある文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、各行において最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えさせることを特徴とする。

【0015】上記目的を達成するため、請求項11の発明は、コンピュータによってソフトウェアのソースコードに識別情報を埋め込むためのソフトウェア識別情報埋め込みプログラムを記録した記録媒体であって、該プログラムはコンピュータに対し、前記識別情報をビット列で表現した透かしビット列に変換させ、前記ソースコードの中で前記透かしビットを埋め込む位置を指定させ、前記指定されたソースコードの位置にある、文字あるいは記号と他の文字あるいは記号の間にある文字記号間スペースと、行の最初の文字あるいは最初の記号が出現するまでに存在する行先頭スペースのうちで、両方のスペースのスペース長（ブランクの個数）、あるいは、いずれかのスペースのスペース長を、前記透かしビット列の各ビットの値に応じて与えさせることを特徴とする。

【0016】(作用)本発明では、ソフトウェアのソースコードに識別情報を埋め込むソフトウェア識別情報埋め込みにおいて、識別情報をビット列で表現した透かしビット列に変換し、文字記号間スペースと行先頭スペースの両方あるいは片方の長さを、その透かしビットの1、0の値に応じて与える。即ち、識別情報を透かしビット列に変換し、このビット列をソースコードのスペース長を用いて表現する。プログラム言語においては、スペース長の大きさはドントケア(don't care)：値が決められ

ていないこと)である場合が多いので、その場合には、プログラム機能上、問題なく識別情報をソフトウェアのソースコードに埋め込むことが可能であり、これによりソフトウェアの識別情報を明らかにできる。

【0017】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を詳細に説明する。

【0018】図1は本発明の一実施形態におけるソフトウェア識別情報埋め込み方法の処理手順を示すフローチャートである。図2は本発明の一実施形態における、その方法を実現するソフトウェア識別情報埋め込みシステムの構成を示すブロック図である。

【0019】図1および図2に示すように、まず、操作部のキーボード等(図示しない)から入力した著作権情報や入手者情報などの識別情報を、識別情報ビット列化部21により、ビット列で表現した透かしビット列に変換する(ステップ1)。

【0020】その際、その識別情報を文字や記号で表現する場合には、A S C I I コードや、E B C D I C コード、J I S コードなどの文字コードに従ってビット列に変換する。識別情報をロゴマークで表現する場合には、ロゴマークを2値パターン(例えば、8×8の2次元メッシュパターン)で表現してから、これを1次元のビット列に変換する。

【0021】次に、埋め込み位置指定部22により、透かしビットを埋め込むソフトウェアのソースコードの位置を行番号、スペース番号などで指定する(ステップ2)。但し、埋め込み位置をいちいち指定する必要がない場合(例えば、先頭行から始めて、文字記号間スペースがあるたびに埋め込む場合)には、図1のステップ2と図2の埋め込み位置指定部22は不要である。

【0022】さらに、ステップ2で指定された行にある文字記号間スペースと行先頭スペースの、両方あるいは

<例示2>

```
第1行 #include <stdio.h>
第2行 #include <math.h>
第3行 #define GAMMA 0.35
第4行 #define DELTA 0.30
第5行 FILE *win;
第6行 FILE *wout;
```

【0028】透かしビットを埋め込むソースコードの行に、文字記号間スペースが複数個ある場合、埋め込みを利用する文字記号間スペースの位置を選択して指定してもよい。例えば、ビット列「0 1 1 0 1 1」をC言語で書かれたソースコードの第1行から第6行までの文字記

<例示3>

```
第1行 #include <stdio.h>
第2行 #include <math.h>
```

片方の長さを、スペース長付与部23により、ステップ1で得られた上記の透かしビットの値に応じて与える(ステップ3)。

【0023】例えば、透かしビットの値が0のときには、スペース長(スペースの両方あるいは片方の長さ)を1とし、透かしビットの値が1のときには、スペース長を2とする。ステップ2で指定された行にスペースがない場合には、何もしないで次の行へとぶ。

【0024】本発明のこのようなソフトウェア識別情報埋め込み方法を適用する対象のソースコードの一例として、C言語のソースコードを、例示1として以下に示す(ここでは、紙面の都合で最初の6行までと101~103行だけを示す)。

【0025】

【表1】

<例示1>

```
第1行 #include <stdio.h>
第2行 #include <math.h>
第3行 #define GAMMA 0.35
第4行 #define DELTA 0.30
第5行 FILE *win;
第6行 FILE *wout;
第101行 tin[i][j][j] = GAMMA;
第102行 tin[i][j][rank[i][j][0]] = 0.9;
第103行 t1[rank[i][j][0]][j] = 1;
```

【0026】透かしビットの値が0のときには、スペース長を1とし、透かしビットの値が1のときには、スペース長を2とする場合、透かしビット列の一部である「0 1 1 0 1 1」をC言語で書かれたソースコードの第1行から第4行までの文字記号間スペースすべてを利用して埋め込む場合には、下記の例示2のようになる。

【0027】

【表2】

```
/* eと<の間にプランク1個 */
/* eと<の間にプランク2個 */
/* eとGの間にプランク2個、
   Aと0の間にプランク1個 */
/* eとDの間にプランク2個、
   Aと0の間にプランク2個 */
```

号間スペースのうち、各行の最初の文字記号間スペースのみを利用して埋め込む場合には、下記の例示3のようになる。

【0029】

【表3】

```
/* eと<の間にプランク1個 */
/* eと<の間にプランク2個 */
```

```

第3行 #define GAMMA 0.35 /* eとGの間にブランク2個 */
第4行 #define DELTA 0.30 /* eとDの間にブランク1個 */
第5行 FILE *win; /* Eと*の間にブランク2個 */
第6行 FILE *wout; /* Eと*の間にブランク2個 */

```

【0030】JIS漢字コードでは1字2バイトなので、例えば、著作者名「特許太郎」は64ビットの透かしビットに変換される。また、ロゴマークを8×8のメッシュパターンで表現した場合にも64ビットの透かしビットに変換される。十分大規模なソースコード（例えば、10万行）のうち、高々64行に透かしビットを埋め込んだ場合、その埋め込み位置を知らずに、その透かしビットを見つけ出すのは非常に困難である。

【0031】透かしビットを埋め込むソースコードの行番号に応じて、スペース長の与え方を変えてよい。例えば、行番号が奇数のときには「透かしビットの値が0のときにはスペース長を1とし、透かしビットの値が1

<例示4>

```

第1行 #include <stdio.h> /* eと<の間にブランク1個 */
第2行 #include <math.h> /* eと<の間にブランク1個 */
第3行 #define GAMMA 0.35 /* eとGの間にブランク2個、
                           Aと0の間にブランク1個 */
第4行 #define DELTA 0.30 /* eとDの間にブランク1個、
                           Aと0の間にブランク1個 */
第5行 FILE *win; /* eと*の間にブランク1個 */
第6行 FILE *wout; /* eと*の間にブランク1個 */

```

【0034】同一の識別情報をソースコードの複数箇所に反復して埋め込んでもよい。こうすることによって、識別情報を何箇所にも書けるので改竄されにくくなる。

【0035】ビット列「011011」をC言語で書かれたソースコードの第1行から第4行までの文字記号間

<例示5>

```

第1行 #include <stdio.h> /* eと<の間にブランク1個 */
第2行 #include <math.h> /* eと<の間にブランク2個 */
第3行 #define GAMMA 0.35 /* eとGの間にブランク2個、
                           Aと0の間にブランク1個 */
第4行 #define DELTA 0.30 /* eとDの間にブランク2個、
                           Aと0の間にブランク2個 */
第5行 FILE *win; /* eと*の間にブランク1個 */
第6行 FILE *wout; /* eと*の間にブランク1個 */
第101行 tin[i][j][j] = GAMMA; /* ]と=の間にブランク1個、
                                 =とGの間にブランク2個 */
第102行 tin[i][j][rank[i][j][0]] = 0.9; /* ]と=の間にブランク2個
                                         =と0の間にブランク1個 */
第103行 t1[rank[i][j][0]][j] = 1; /* ]と=の間にブランク2個
                                         =と1の間にブランク2個 */

```

【0037】更に、透かしビット列の一部である「011」をC言語で書かれたソースコードの第101行から第103行までの行先頭スペース長で表現する場合において、透かしビットが0のときにはスペース長0、透か

<例示6>

のときにはスペース長を2」とし、行番号が偶数のときには「透かしビットの値が0のときにはスペース長を2とし、透かしビットの値が1のときにはスペース長を1」とする。

【0032】この行番号の奇・偶数に応じて変える方法を用いて、透かしビット列「011011」をC言語に書かれたソースコードの第1行から第4行までの文字記号間スペースのすべてを利用して埋め込む場合には、下記の例示4のようになる。

【0033】

【表4】

```

/* eと<の間にブランク1個 */
/* eと<の間にブランク1個 */
/* eとGの間にブランク2個、
   Aと0の間にブランク1個 */
/* eとDの間にブランク1個、
   Aと0の間にブランク1個 */

```

スペースすべてと、第101行から第103行までの文字記号間スペースすべてを利用して、2回埋め込む場合には、下記の例示5のようになる。

【0036】

【表5】

```

/* eと<の間にブランク1個 */
/* eと<の間にブランク2個 */
/* eとGの間にブランク2個、
   Aと0の間にブランク1個 */
/* eとDの間にブランク2個、
   Aと0の間にブランク2個 */

```

シビットが1のときにはスペース長1としたときには、下記の例示6のようになる。

【0038】

【表6】

```

第101 行 tin[i][j][j] = GAMMA ; /* t の前にブランクなし */
第102 行 tin[i][j][rank[i][j][0]] = 0.9; /* t の前にブランク 1 個 */
第103 行 tl[rank[i][j][0]][j] = 1; /* t の前にブランク 1 個 */

```

【0039】また、文字記号間スペースと行先頭スペースの両方を利用して埋め込むことも可能である。

【0040】透かしビットを埋め込む場所を指定する方法としては、行単位の指定のほかに、プログラム機能単位あるいはコメント文単位で指定する方法や、機能単位、コメント単位と行単位を併用して指定する方法にも適用可能である。また、埋め込み開始位置を記号で指定する方法（例えば、C言語では、n番目の「=」から始める）にも適用可能である。

【0041】ソースコードの規模に応じて、埋め込む識別情報の大きさを変える方法も有効である。例えば、大規模なソースコード（例えば、10万行）の場合には、識別情報として、作者のフルネームと所属組織、連絡先をすべて（大きな識別情報）埋め込むが、小規模な（例えば、1000行）場合には作者のファミリーネームをアルファベットで（小さな識別情報）を埋め込む。大きな識別情報と小さな識別情報の両者を別々の箇所に埋め込むことも好ましい。

【0042】識別情報埋め込みに関する具体的方法（埋め込みに利用したスペースの種類、埋め込んだ行番号、反復して埋め込んだか否か、透かしビット値とスペース長の対応関係、埋め込みに利用したスペースの位置、行番号に応じてスペース長の与え方を変えたか否かなど）を暗号化して所定の記憶手段に記憶し、管理することで秘匿性を高めることができる。この場合、埋め込み位置

（行番号、スペース番号など）を暗号化鍵（キー）で指定し、透かし情報を得る場合には復号鍵を用いて埋め込み位置を特定する。

【0043】なお、本実施形態においては、C言語を例示したが、本発明はC言語に限らず、ソースコードに文字記号間スペースあるいは行先頭スペースを有する他の言語（プログラミング言語、システム設計言語、動作記述言語、WWW用記述言語など）にも同様にして適用可能であることは明らかである。

【0044】本発明は、上述のように、横方向のスペース長を利用して透かしビットを埋め込む方法に関するものであるが、本発明の基本的な考えは、縦方向の行間隔

を利用して透かしビットを埋め込む方法にも適用可能である。

【0045】（他の実施形態）なお、本発明の目的は、前述した実施の形態の機能を実現するソフトウェアのプログラムコードを記録した記録媒体（記憶媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記録媒体に格納されたプログラムコードを読み出し、実行することによっても、達成されることは言うまでもない。この場合、記録媒体から読み出されたプログラムコード自分が前述した実施の形態の機能を実現することになり、そのプログラムコードを記録した記録媒体は本発明を構成することになる。そのプログラムコードを記録し、またテープ等の変数データを記録する記録媒体としては、例えばフロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROMなどを用いことができる。

【0046】
【発明の効果】以上説明したように、本発明では、識別情報を透かしビット列に変換し、この透かしビット列をソースコードのスペース長を用いて表現するようにしたので、プログラム機能上、問題なく識別情報を埋め込むことが可能であり、ソフトウェアの識別情報を明らかにでき、これにより著作権保護機能を向上させることができる。

【図面の簡単な説明】

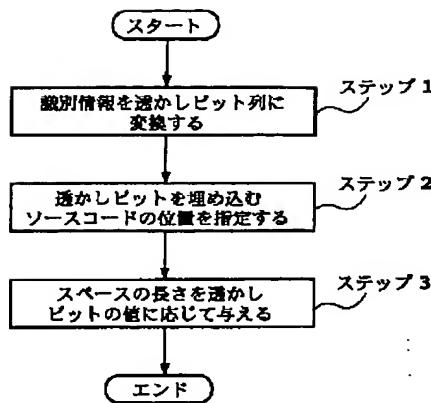
【図1】本発明の一実施形態におけるソフトウェア識別情報埋め込み方法の処理手順を示すフローチャートである。

【図2】本発明の一実施形態における、図1の方法を実現するソフトウェア識別情報埋め込みシステムの構成を示すブロック図である。

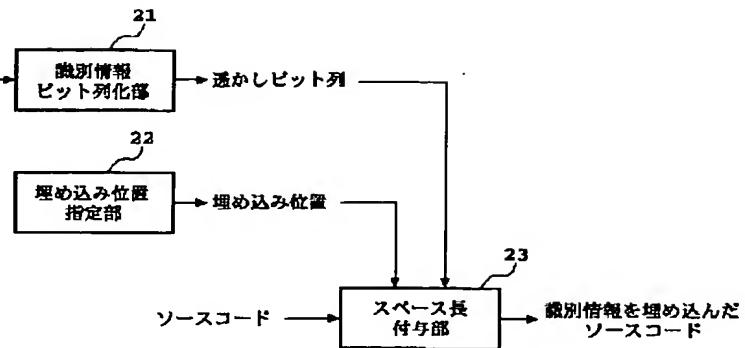
【符号の説明】

- 2 1 識別情報ビット列化部
- 2 2 埋め込み位置指定部
- 2 3 スペース長付与部

【図1】



【図2】



フロントページの続き

(51) Int. Cl. 7

H 04 N 7/08
7/081

識別記号

F I
H 04 N 7/08マーク (参考)
Z 5 J 1 0 4
9 A 0 0 1

F ターム (参考) 5B009 VA00
 5B057 CE08 CG07
 5B076 FB02
 5C063 CA29 CA36 CA40 DA03 DA07
 DA13
 5C076 AA14 BA06
 5J104 AA14 AA15 NA05 NA27
 9A001 DD01 EE03 LL01